

CLAIMS

What is claimed is:

1. A flexible interface employing a client program interconnected to a client server which, in turn, is operably networked to a workflow server running a workflow management system having workflow-type functionality including a set of predefined process templates defining a set of tasks and being capable of monitoring workflow, assigning tasks to users, and allowing users to initiate instances of a process from one of the set of predefined process templates, the interface comprising:

at least one first workflow platform-dependent object for accessing the workflow functionality;

at least one second workflow platform-independent object for providing data to the at least one first workflow platform-dependent object;

a set of predefined protocol user interface pages comprising at least one control page with said predefined protocol having at least one invocation to the at least one second object embedded therein; and

at least one server program callable by the at least one control page and adapted to invoke at least one of the first workflow platform-dependent object and the at least one second workflow platform-independent object;

wherein, when the functionality of the workflow management system is to be accessed via the set of predefined protocol user interface pages, the at least one control page calls the at least one server program which, in turn, invokes at least one of the first and second objects to promote data translation and exchange between the client program and the workflow management system.

2. The flexible interface of claim 1 wherein the server program further comprises a Log On servlet for receiving a user identification variable and a password variable from one of the set of predefined protocol pages and invoking at least one of the first and second objects to authenticate the user identification variable with the workflow management system.

3. The flexible interface of claim 2 wherein the server program further comprises an Activity servlet for receiving a work item identification and a target user interface address from the at least one control page, and invoking at least one of the first and second objects to record lock a task in the workflow management system corresponding to the work item identification and permit exclusive access by a user to the task identified by the work item identification.

4. The flexible interface of claim 3 wherein the Activity servlet further comprises a redirection command for navigating the user to a predefined protocol page represented by the target user interface address for performing work on the task identified by the work item identification.

5. The flexible interface of claim 4 wherein the server program further comprises a Check In servlet for receiving a work item identification from one of an activity user interface page and the at least one control page, and invoking at least one of the first and second objects to release a record lock on a task in the workflow management system corresponding to the work item identification and terminate any exclusive access by a user to the task identified by the work item identification.

6. The flexible interface of claim 5 wherein the Check In servlet further receives task-specific data from a requesting page and further comprises a database update command for updating the task identified by the work item identification with the task-specific

data.

7. The flexible interface of claim 6 wherein the server program further comprises a New Instance servlet for receiving a predefined process template identification from the at least one control page, and invoking at least one of the first and second objects to initiate a new instance of a process template in the workflow management system corresponding to the process template identification.

8. The flexible interface of claim 7 wherein the New Instance servlet determines whether input data is needed to initiate the new instance of the predefined process template and the interface further comprises a user interface page wherein the New Instance servlet redirects the user to the user interface page to further receive input data to properly initiate the new instance of the process template with the input data.

9. The flexible interface of claim 8 and further comprising at least one predefined protocol user interface page adapted to receive at least one data variable from the user and to call the at least one server-based applet.

10. The flexible interface of claim 9 wherein the user interface page further comprises at least one <FORM> tag having at least one input element for receiving data from the user.

11. The flexible interface of claim 10 wherein the user interface page is adapted to provide at least one data variable to initiate a process template into a running process that requires the data entry for instantiation.

12. The flexible interface of claim 11 wherein the user interface page is adapted to provide at least one data variable to complete a task from a previously initiated process that requires the

data entry for completion.

13. The flexible interface of claim 12 wherein the first and second objects are Java classes.

5

14. The flexible interface of claim 13 wherein the first object is implemented in terms of the workflow-type functionality of the workflow management system.

10 15. The flexible interface of claim 14 wherein the at least one control page comprises a page selected from a group consisting of a worklist page, a process instance page, a process template page, a work item control page, a filter control page, and a notifications page.

16. The flexible interface of claim 14 wherein the client program is a web browser and the client server is a web server.

17. The flexible interface of claim 1 wherein the server program further comprises an Activity servlet for receiving a work item identification and a target user interface address from the at least one control page, and invoking at least one of the first and second objects to record lock a task in the workflow management system corresponding to the work item identification and permit exclusive access by a user to the task identified by the work item identification.

18. The flexible interface of claim 17 wherein the Activity servlet further comprises a redirection command for navigating the user to a predefined protocol page represented by the target user interface address for performing work on the task identified by the work item identification.

19. The flexible interface of claim 1 wherein the server program further comprises a Check In servlet for receiving a work item identification from the at least one control page, and invoking at least one of the first and second objects to release a record lock on a task in the workflow management system corresponding to the work item identification and terminate any exclusive access by a user to the task identified by the work item identification.

20. The flexible interface of claim 19 wherein the Check In servlet further receives task-specific data from a requesting page and further comprises a database update command for updating the task identified by the work item identification with the task-specific data.

21. The flexible interface of claim 1 wherein the server program further comprises a New Instance servlet for receiving a predefined process template identification from the at least one control page, and invoking at least one of the first and second objects to initiate a new instance of a process template in the workflow management system corresponding to the process template identification.

22. The flexible interface of claim 21 wherein the New Instance servlet determines whether input data is needed to initiate the new instance of the predefined process template and the interface further comprises a user interface page wherein the New Instance servlet redirects the user to the user interface page to further receive input data to properly initiate the new instance of the process template with the input data.

23. The flexible interface of claim 1 and further comprising at least one predefined protocol user interface page adapted to receive at least one data variable from the user and to call the at least one server-based applet.

24. The flexible interface of claim 23 wherein the user interface page further comprises at least one <FORM> tag having at least one input element for receiving data from the user.

25. The flexible interface of claim 24 wherein the user interface page is adapted to provide at least one data variable to initiate a process template into a running process that requires the data entry for instantiation.

26. The flexible interface of claim 25 wherein the user interface page is adapted to provide at least one data variable to complete a task from a previously initiated process that requires the data entry for completion.

27. The flexible interface of claim 1 wherein the first and second objects are Java classes.

28. The flexible interface of claim 1 wherein the Java class comprises a Java interface class.

29. The flexible interface of claim 1 wherein the at least one control page comprises a page selected from a group consisting of a worklist page, a process instance page, a process template page, a work item control page, a filter control page, and a notifications page.

30. The flexible interface of claim 1 wherein the client program is a web browser and the client server is a web server.

31. A method for employing a flexible interface of a client program interconnected to a client server to access the functionality of a workflow management system operably networked to a workflow server, the functionality including a set of predefined process templates defining a set of tasks and being capable of monitoring workflow, assigning tasks to users, and allowing users to initiate instances of a process from one of the set of predefined process templates, the method comprising the steps of:

configuring a set of predefined protocol user interface pages comprising at least one control page with said predefined protocol having at least one server-side script embedded therein;

pointing at least one first workflow platform-dependent object to access the workflow functionality;

interfacing at least one second workflow platform-independent object with the at least one first workflow platform-dependent object; and

calling at least one server program with the at least one control page which thereby invokes at least one of the first workflow platform-dependent object and the at least one second workflow platform-independent object;

wherein, when the functionality of the workflow management system is to be accessed via the set of predefined protocol user interface pages, the at least one control page calls the at least one server program which, in turn, invokes at least one of the first and second objects to promote data translation and exchange between the client program and the workflow management system.

32. The method of claim 31 and further comprising the step of logging a user on to the workflow management system via one of the set of predefined protocol pages, receiving a user identification variable and a password variable therefrom, and invoking at least one of the first and second objects to authenticate the user identification variable with the workflow management system.

33. The method of claim 32 and further comprising the step of receiving a work item identification and a target user interface address from the at least one control page; and invoking at least one of the first and second objects to record lock a task in the workflow management system corresponding to the work item identification.

34. The method of claim 33 and further comprising the step of issuing a redirection command to a predefined protocol page represented by the target user interface address for performing work on the task identified by the work item identification.

35. The method of claim 34 and further comprising the step of receiving a work item identification from the at least one control page and invoking at least one of the first and second objects to release a record lock on a task in the workflow management system corresponding to the work item identification and terminate any exclusive access by a user to the task identified by the work item identification.

36. The method of claim 35 and further comprising the step of receiving task-specific data from a requesting page and updating the task identified by the work item identification with the task-specific data in the workflow management system.

37. The method of claim 36 and further comprising the step of receiving a predefined process template identification from the at least one control page, and invoking at least one of the first and second objects to initiate a new instance of a process template in the workflow management system corresponding to the process template identification.

38. The method of claim 37 and further comprising the step of determining whether input data is needed to initiate the new instance of the predefined process template.

39. The method of claim 38 and further comprising the step of redirecting the user to a user interface page to receive required input data to properly initiate the new instance of the process template with the input data if the determining step results in a determination that data is needed to initiate the process instance.

40. The method of claim 39 and further comprising the step of redirecting the user from the user interface page back to the New Instance servlet to update the workflow management system with the received input data to initiate the process instance.

41. The method of claim 40 and further comprising at least one predefined protocol user interface page adapted to receive at least one data variable from the user and to call the at least one server-based applet.

42. The method of claim 41 wherein the user interface page further comprises at least one <FORM> tag having at least one input element for receiving data from the user.

43. The method of claim 42 wherein the user interface page is adapted to provide at least one data variable to initiate a process template into a running process that requires the data entry for instantiation.

44. The method of claim 43 wherein the user interface page is adapted to provide at least one data variable to complete a task from a previously initiated process that requires the data entry for completion.

45. The method of claim 44 wherein the first and second objects are Java classes.

46. The method of claim 45 wherein the Java class comprises a Java interface class.

47. The method of claim 45 wherein the at least one control page comprises a page selected from a group consisting of a worklist page, a process instance page, a process template page, a work item control page, a filter control page, and a notifications page.

48. The method of claim 45 wherein the client program is a web browser and the client server is a web server.

49. The method of claim 31 and further comprising the step of receiving a work item identification and a target user interface address from the at least one control page; and invoking at least one of the first and second objects to record lock a task in the workflow management system corresponding to the work item identification.

50. The method of claim 49 and further comprising the step of issuing a redirection command to a predefined protocol page represented by the target user interface address for performing work on the task identified by the work item identification.

51. The method of claim 31 and further comprising the step of receiving a work item identification from the at least one control page and invoking at least one of the first and second objects to release a record lock on a task in the workflow management system corresponding to the work item identification and terminate any exclusive access by a user to the task identified by the work item identification.

52. The method of claim 51 and further comprising the step of receiving task-specific data from a requesting page and updating the task identified by the work item identification with the task-specific data in the workflow management system.

53. The method of claim 31 and further comprising the step of receiving a predefined process template identification from the at least one control page, and invoking at least one of the first and second objects to initiate a new instance of a process template in the workflow management system corresponding to the process template identification.

54. The method of claim 53 and further comprising the step of determining whether input data is needed to initiate the new instance of the predefined process template.

55. The method of claim 54 and further comprising the step of redirecting the user to a user interface page to receive required input data to properly initiate the new instance of the process template with the input data if the determining step results in a determination that data is needed to initiate the process instance.

56. The method of claim 55 and further comprising the step of redirecting the user from the user interface page back to the New Instance servlet to update the workflow management system with the received input data to initiate the process instance.

57. The method of claim 31 and further comprising at least one predefined protocol user interface page adapted to receive at least one data variable from the user and to call the at least one server-based applet.

58. The method of claim 57 wherein the user interface page further comprises at least one <FORM> tag having at least one input element for receiving data from the user.

59. The method of claim 57 wherein the at least one user interface page is adapted to provide at least one data variable to initiate a process template into a running process that requires the data entry for instantiation.

60. The method of claim 57 wherein the user interface page is adapted to provide at least one data variable to complete a task from a previously initiated process that requires the data entry for completion.

61. The method of claim 31 wherein the first and second objects are Java classes.

62. The method of claim 61 wherein the Java class comprises a Java interface class.

63. The method of claim 31 wherein the at least one control page comprises a page selected from a group consisting of a worklist page, a process instance page, a process template page, a work item control page, a filter control page, and a notifications page.

64. The method of claim 31 wherein the client program is a web browser and the client server is a web server.

65. A method for organizing and locating and navigating users within a flexible interface of a client program interconnected to a client server having a predefined interface root directory path to access the functionality of a workflow management system operably
5 networked to a workflow server, the functionality including a set of predefined process templates each having a unique process identifier and defining a set of tasks and being capable of monitoring workflow, assigning tasks to users each having a unique task identifier, and allowing users to initiate instances of a process from one of the set
10 of predefined process templates, the method comprising the steps of:

creating at least one predefined protocol process activity page relating to a process and named for the unique process identifier;

locating the at least one predefined protocol process activity page in the predefined interface root directory path;

5 creating a process directory beneath the predefined interface root directory path for the process and named for the unique identifier thereof;

creating at least one predefined protocol user interface page within the created process directory in a predetermined protocol relating to a task assignable within the process named for the task
20 unique identifier if the process requires input on any of its assignable activities;

locating the at least one predefined protocol user interface page in the created directory within the predefined interface root
25 directory path; and

whereby the predefined protocol process activity page can be automatically located by the interface within the predefined interface root directory path of the client server by only knowing the process unique identifier and the at least one predefined protocol user
30 interface page can be located in the created directory within the predefined interface root directory path by knowing only the task unique identifier.

00220110200
09704916
09704916

66. The method of claim 65 and further comprising the step of embedding a form within the at least one user interface page in the predefined protocol configured so as to provide any required data to the assignable task into the workflow management system.

5

67. The method of claim 66 wherein the form contains input prompts configured so as to provide specific data in a machine-readable format to the workflow management system.

10 68. The method of claim 67 and further comprising the step of embedding a hidden field on the at least one user interface page containing the unique process identifier for cross-referencing the data within the at least one user interface page with the workflow management system.

69. The method of claim 68 wherein the predefined protocol comprises at least one of HTML and javascript.

70. The method of claim 66 and further comprising the step of defining a programming object for use as an input container for delivering data entered by a user on the at least one user interface page.

05704916
20
002015